Examiner-Initiated Interview Summary	Application No.	Applicant(s)	
	10/722,480	BOON ET AL.	
	Examiner	Art Unit	
	Kenneth Tang	2195	

All Participants:

(1) *Kenneth Tang*.

(2) *John C. Gorecki*.

Status of Application: *Amendment after non-final*

(3) _____.

(4) _____.

Date of Interview: _____

Time: *5:30pm*

Type of Interview:
- ☒ Telephonic
- ☐ Video Conference
- ☐ Personal (Copy given to: ☐ Applicant ☐ Applicant's representative)

Exhibit Shown or Demonstrated: ☐ Yes ☐ No
If Yes, provide a brief description:

Part I.

Rejection(s) discussed:
NA

Claims discussed:
1, 12, 14-15, and 23

Prior art documents discussed:
NA

Part II.

SUBSTANCE OF INTERVIEW DESCRIBING THE GENERAL NATURE OF WHAT WAS DISCUSSED:
See Continuation Sheet

Part III.

☒ It is not necessary for applicant to provide a separate record of the substance of the interview, since the interview directly resulted in the allowance of the application. The examiner will provide a written summary of the substance of the interview in the Notice of Allowability.

☐ It is not necessary for applicant to provide a separate record of the substance of the interview, since the interview did not result in resolution of all issues. A brief summary by the examiner appears in Part II above.

Kenneth Tang
(Examiner/SPE Signature)

(Applicant/Applicant's Representative Signature – if appropriate)

Continuation of Substance of Interview including description of the general nature of what was discussed: It was agreed to ammend the Specification to ensure clarity that the claimed computer readable medium was an example of the software medium in the Specification, which is related to the disk, tape, chip or a random access memory. Improper claim numbering were corrected by the Examiner and the term "scheduling" in the preamble was amended to "calculating" to further improve clarity.

[File 348] **EUROPEAN PATENTS** 1978-2007/ 200751

**File 348: For important information about IPCR/8 and forthcoming changes to the IC= index, see HELP NEWSIPCR.*

[File 349] **PCT FULLTEXT** 1979-2007/UB=20071227UT=20071120

**File 349: For important information about IPCR/8 and forthcoming changes to the IC= index, see HELP NEWSIPCR.*

; d s

Set	Items	Postings	Description
S1	369899	6671970	S CPU OR CPUS OR PROCESSOR? ? OR MICROPROCESSOR? ? OR CENTRAL()PROCESSING()UNIT? ?
S2	27910	139232	S S1(3N)(USAGE OR UTILIZ? OR UTILIS? OR COST? ? OR LOAD???)
S3	1524309	24139337	S TASK? ? OR PROCESS?? OR JOB? ?
S4	53560	291516	S S3(3N)(CLASS?? OR CLASSIFICATION? ? OR CATEGOR? OR GROUP??? OR SUBCLASS?? OR SUBGROUP? OR SUBCATEGOR?)
S5	101582	1111395	S HIERARCH? OR TREE? ? OR BTREE? ?
S6	1175368	14329075	S WEIGH???? OR IMPORTANCE OR PRIORIT?
S7	112	550	S S2(3N)(MULTIPLY??? OR MULTIPLIE? ? OR MULTIPLICATION OR CO()EFFICIENT? ? OR COEFFICIENT? ?)
S8	8	198	S S7(100N)S6
S9	1	5	S S7(100N)S4
S10	11	136	S S2(50N)S4(50N)S5
S11	46	557	S S2(50N)S4(50N)S6
S12	56	691	S S10:S11
S13	48	722	S S12 AND IC=G06F
S14	47	707	S S13 NOT S8:S9
S15	5	42	S S7(100N)S5
S16	5	42	S S15 NOT (S8 OR S9)

8/3K/6 (Item 2 from file: 349) Links
PCT FULLTEXT

01201922

METHOD AND SYSTEM FOR MONITORING A COMPUTER SYSTEM FOR DYNAMIC RECONFIGURATION

PROCEDE ET SYSTEME POUR LE CONTROLE DE SYSTEME INFORMATIQUE AUX FINS DE RECONFIGURATION DYNAMIQUE

Patent Applicant/Patent Assignee:

E. **COMPUTER ASSOCIATES THINK INC**; One Computer Associates Plaza, Islandia, NY 11749
; ;

F. **ESFAHANY Kouros H**; 40 Briarwood Drive, Huntington, NY 11743
US; US(Residence); US(Nationality)
(Designated only for: US)

Patent Applicant/Inventor:

G. **ESFAHANY Kouros H**
40 Briarwood Drive, Huntington, NY 11743; US; US(Residence); US(Nationality); (Designated only for: US)

Legal Representative:

H. **JAWORSKI Richard F(agent)**
Cooper & Dunham LLP, 1185 Avenue of the Americas, New York, NY 10036; US;

	Country	Number	Kind	Date
Patent	WO	200508497	A2-A3	20050127
Application	WO	2004US22149		20040709
Priorities	US	2003486829		20030711

Designated States: (All protection types applied unless otherwise stated - for applications 2004+)
AE; AG; AL; AM; AT; AU; AZ; BA; BB; BG;
BR; BW; BY; BZ; CA; CH; CN; CO; CR; CU;
CZ; DE; DK; DM; DZ; EC; EE; EG; ES; FI;
GB; GD; GE; GH; GM; HR; HU; ID; IL; IN;
IS; JP; KE; KG; KP; KR; KZ; LC; LK; LR;
LS; LT; LU; LV; MA; MD; MG; MK; MN; MW;
MX; MZ; NA; NI; NO; NZ; OM; PG; PH; PL;
PT; RO; RU; SC; SD; SE; SG; SK; SL; SY;
TJ; TM; TN; TR; TT; TZ; UA; UG; US; UZ;
VC; VN; YU; ZA; ZM; ZW;

bad date?
FH

[EP] AT; BE; BG; CH; CY; CZ; DE; DK; EE; ES;
FI; FR; GB; GR; HU; IE; IT; LU; MC; NL;
PL; PT; RO; SE; SI; SK; TR;

[OA] BF; BJ; CF; CG; CI; CM; GA; GN; GQ; GW;
ML; MR; NE; SN; TD; TG;

[AP] BW; GH; GM; KE; LS; MW; MZ; NA; SD; SL;
SZ; TZ; UG; ZM; ZW;

[EA] AM; AZ; BY; KG; KZ; MD; RU; TJ; TM;

Publication Language: English
Filing Language: English
Fulltext word count: 6411

Detailed Description:

...the analyzing step S24, the present status of a domain is determined based on the **weighted** sum of the attribute values. More specifically, the current status (DS) of a domain may... ...i - number of the domain
j - number of collections of i-th domain
K,pu - **weight coefficient** for **CPU usage**, the **CPU** attribute value
K,,, - **weight** coefficient for Memory usage, the memory attribute value
Kia **weight** coefficient for Load Average, the load average attribute value
Cij current CPU attribute value
Mij...

[File 2] **INSPEC** 1898-2007/Dec W2

[File 6] **NTIS** 1964-2008/Jan W2

[File 8] **Ei Compendex(R)** 1884-2007/Dec W3

[File 23] **CSA Technology Research Database** 1963-2007/Nov

[File 34] **SciSearch(R) Cited Ref Sci** 1990-2007/Dec W5

[File 35] **Dissertation Abs Online** 1861-2007/Oct

[File 65] **Inside Conferences** 1993-2007/Dec 31

[File 95] **TEME-Technology & Management** 1989-2007/Dec W3

[File 99] **Wilson Appl. Sci & Tech Abs** 1983-2007/Oct

[File 144] **Pascal** 1973-2007/Dec W2

[File 256] **TecInfoSource** 82-2007/Jul

[File 434] **SciSearch(R) Cited Ref Sci** 1974-1989/Dec

[File 239] **Mathsci** 1940-2007/Dec


```
; d s
Set   Items   Postings   Description
S1    635217    1376772   S CPU OR CPUS OR PROCESSOR? ? OR MICROPROCESSOR? ? OR
CENTRAL()PROCESSING()UNIT? ?
S2     24713      60693   S S1(3N)(USAGE OR UTILIZ? OR UTILIS? OR COST? ? OR LOAD???)
S3   9415320   17375717   S TASK? ? OR PROCESS?? OR JOB? ?
S4    102399     260329   S S3(3N)(CLASS?? OR CLASSIFICATION? ? OR CATEGOR? OR GROUP??? OR
SUBCLASS?? OR SUBGROUP? OR SUBCATEGOR?)
S5    963032    2242445   S HIERARCH? OR TREE? ? OR BTREE? ?
S6   3189243    5086908   S WEIGH???? OR IMPORTANCE OR PRIORIT?
S7        59        200   S S2(3N)(MULTIPLY??? OR MULTIPLIE? ? OR MULTIPLICATION OR
CO()EFFICIENT? ? OR COEFFICIENT? ?)
S8         3         26   S S7 AND S6
S9         0          0   S S7 AND S4
S10       38        366   S S2 AND S4 AND S5
S11       38        366   S S10 NOT S8
S12       21        205   RD (unique items)
S13       17        149   S S12 NOT PY=2004:2008
```

S14	3	26	S S7 AND S6
S15	39	234	S S7 AND S3
S16	36	212	S S14:S15 NOT (S8 OR S13)
S17	26	159	RD (unique items)
S18	22	137	S S17 NOT PY=2004:2008

8/5/1 (Item 1 from file: 2) Links

Fulltext available through: USPTO Full Text Retrieval Options STIC Full Text Retrieval Options

03637510 **INSPEC Abstract Number:** C86024909

Title: Analysis of the efficiency of multiprocessor systems with a common computing resource, under various disciplines of its allocation

Author Zabolotnyi, A.A.; Nedzel'skii, D.A.

Journal: Avtomatika i Vychislitel'naya Tekhnika vol.18, no.6 p. 60-4

Publication Date: 1984 **Country of Publication:** USSR
CODEN: AVYTAK **ISSN:** 0132-4160
Translated in: Automatic Control and Computer Sciences vol.18, no.6 **p.** 55-9
Publication Date: 1984 **Country of Publication:** USA
CODEN: ACCSCE **ISSN:** 0146-4116
U.S. Copyright Clearance Center Code: 0146-4116/84/$20.00

Language: English **Document Type:** Journal Paper (JP)
Treatment: Theoretical (T)

Abstract: The effect on the system efficiency of the discipline with which common computing resources (vector processors) are furnished to control processors in a multiprocessor computer system of type PS-3000 is investigated. The analysis procedure is demonstrated using the example of a system with two control processors. Three disciplines for furnishing the common computing resource are considered: equiprobable choice, dynamic **priority** and static **priority**. Approximate analytic expressions are obtained for the **processor utilisation coefficients** under exponential distribution laws from the request generation and execution times. It is shown that the use of dynamic **priority** increases the system performance by 2-4% as compared to other disciplines. Use of static **priority** ensures a quasilimiting utilisation factor for a control processor. It is concluded that, in multiprocessor systems with common computing resource, it is advisable to implement a combined resource-furnishing discipline: static **priority** when the **priorities** of the jobs are different; and dynamic **priority** when they are equal. Recommendations are given on the choice of discipline with a system of nonuniform flows of requests for the common resource. (4 Refs)

Subfile: C
Descriptors: multiprocessing systems; performance evaluation; scheduling
Identifiers: resource allocation disciplines; scheduling; multiprocessor systems; common computing resource; vector processors; control processors; PS-3000; equiprobable choice; dynamic **priority**; static **priority**; **processor utilisation coefficients**; exponential distribution laws; system performance; quasilimiting utilisation factor; nonuniform flows
Class Codes: C5440 (Multiprocessor systems and techniques); C5470 (Performance evaluation and testing); C6150J (Operating systems)

8/5/2 (Item 1 from file: 8) Links

Fulltext available through: USPTO Full Text Retrieval Options STIC Full Text Retrieval Options

Ei Compendex(R)

07105104 **E.I. No:** EIP95032616733

Title: Neural network load-flow

Author: Nguyen, T.T.

Corporate Source: Univ of Western Australia, Nedlands, Aust

Source: IEE Proceedings Generation, Transmission and Distribution v 142 n 1 Jan 1995. p 51-58

Publication Year: 1995

CODEN: IGTDE2 **ISSN:** 1350-2360

Language: English

Document Type: JA; (Journal Article) **Treatment:** A; (Applications); T; (Theoretical)

Journal Announcement: 9505W2

Abstract: The development of a neural network architecture, which implements the Newton-Raphson algorithm for solving the set of nonlinear equations of power system load-flow analysis, is reported. The context is that of online network analysis in energy management systems, with particular reference to the optimal power-flow function. A principal feature of the extensive parallel processing capability of the architecture is that the computing time of load-flow analysis is independent of the number of nodes in the power network. The architecture gives solutions which are identical with those from a standard sequential processor load-flow program. 6 Refs.

Descriptors: *Neural networks; Electric load flow; Algorithms; Online systems; Electric load management; Problem solving; Linearization; Iterative methods; Functions; Matrix algebra

Identifiers: Newton Raphson algorithms; Online network analysis; Optimal power flow function; Minimization problem; Linearized equation system; Objective functions; **Weighting coefficients**; Jacobian matrix; Sequential **processor load** flow program

Classification Codes:

723.1.1 (Computer Programming Languages)

723.4 (Artificial Intelligence); 706.1 (Electric Power Systems); 921.6 (Numerical Methods); 921.1 (Algebra); 722.4 (Digital Computers & Systems); 723.1 (Computer Programming)

723 (Computer Software); 706 (Electric Transmission & Distribution); 921 (Applied Mathematics); 722 (Computer Hardware)

72 (COMPUTERS & DATA PROCESSING); 70 (ELECTRICAL ENGINEERING); 92 (ENGINEERING MATHEMATICS)

13/5/4 (Item 4 from file: 2) Links

Fulltext available through: USPTO Full Text Retrieval Options STIC Full Text Retrieval Options

INSPEC

06470448 **INSPEC Abstract Number:** C9702-6150J-012

Title: CPU inheritance scheduling

Author Ford, B.; Susarla, S.

Author Affiliation: Dept. of Comput. Sci., Utah Univ., Salt Lake City, UT, USA

Journal: Operating Systems Review **Conference Title:** Oper. Syst. Rev. (USA) vol.30, spec. issue. p. 91-105

Publisher: ACM ,

Publication Date: 1996 **Country of Publication:** USA

CODEN: OSRED8 **ISSN:** 0163-5980

SICI: 0163-5980(1996)30L.91:IS;1-D

Material Identity Number: O043-96007

Conference Title: Second USENIX Symposium on Operating Systems Design and Implementation (OSDI)

Conference Date: 28-31 Oct. 1996 **Conference Location:** Seattle, WA, USA

Language: English **Document Type:** Conference Paper (PA); Journal Paper (JP)

Treatment: Practical (P)

Abstract: Traditional processor scheduling mechanisms in operating systems (OSs) are fairly rigid, often supporting only one fixed scheduling policy, or, at most, a few "scheduling classes" whose implementations are closely tied together in the OS kernel. This paper presents CPU inheritance scheduling, a novel processor scheduling framework in which arbitrary threads can act as schedulers for other threads. Widely different scheduling policies can be implemented under the framework, and many different policies can coexist in a single system, providing much greater scheduling flexibility. Modular, **hierarchical** control can be provided over the **processor utilization** of arbitrary administrative domains, such as **processes**, **jobs**, users and **groups**, and the CPU resources consumed can be accounted for and attributed accurately. Applications, as well as the OS, can implement customized local scheduling policies; the framework ensures that all the different policies work together logically and predictably. As a side effect, the framework also cleanly addresses priority inversion by providing a generalized form of priority inheritance that automatically works within and among diverse scheduling policies. CPU inheritance scheduling extends naturally to multiprocessors, and supports processor management techniques such as processor affinity and scheduler activations. We show that this flexibility can be provided with acceptable overhead in typical environments, depending on factors such as context switch speed and frequency. (32 Refs)

Subfile: C

Descriptors: operating system kernels; processor scheduling

Identifiers: CPU inheritance scheduling; processor scheduling mechanisms; operating system kernel; threads; priority inversion; scheduling flexibility; modular **hierarchical** control; **processor utilization**; administrative domains; CPU resource consumption; customized local scheduling policies; multiprocessors; processor management techniques; processor affinity; scheduler activations; overhead; context switch speed; context switch frequency

Class Codes: C6150J (Operating systems); C6150N (Distributed systems software)

13/5/9 (Item 9 from file: 2) Links
INSPEC

04287770 **INSPEC Abstract Number:** C89008448
Title: Hierarchical workload allocation for distributed systems
Author Bowen, N.S.; Nikolaou, C.N.; Ghafoor, A.
Author Affiliation: IBM Thomas J. Watson Res. Center, Yorktown Heights, NY, USA
Conference Title: Proceedings of the 1988 International Conference on Parallel Processing p. 102-9 vol.2
Editor(s): Briggs, F.A.
Publisher: Pennsylvania State Univ , University Park, PA, USA
Publication Date: 1988 **Country of Publication:** USA 3 vol. (xii+461+x+262+xiii+311) pp.
ISBN: 0 271 00654 4
Conference Sponsor: Pennsylvania State Univ
Conference Date: 15-19 Aug. 1988 **Conference Location:** University Park, PA, USA
Availability: Penn State Press, University Park, PA, USA

Language: English **Document Type:** Conference Paper (PA)
Treatment: Practical (P)
Abstract: An efficient **hierarchical** clustering and allocation algorithm is proposed and evaluated that drastically reduces the interprocess communication cost while observing low and upper bounds of **utilization** for the individual **processors**. The algorithm with branch-and-bound-type algorithms that can produce allocations with minimal communication costs is compared, and a very encouraging time complexity/suboptimality tradeoff is shown in favor of present algorithm, at least for a **class** of **process** clusters and their random combinations, which is believed to occur naturally in distributed applications. Present heuristic allocation is well suited for a changing environment, where processors may fail or be added to the system and where the workload patterns may change unpredictably and/or periodically. (6 Refs)
Subfile: C
Descriptors: distributed databases
Identifiers: hierarchical workload allocation; distributed systems; clustering; allocation algorithm; interprocess communication; upper bounds; branch-and-bound-type algorithms; minimal communication costs; time complexity; suboptimality; process clusters
Class Codes: C6160B (Distributed DBMS)

[File 347] **JAPIO** Dec 1976-2007/Jun(Updated 070926)

[File 350] **Derwent WPIX** 1963-2007/UD=200801

**File 350: English-language translations of Chinese Utility Model registrations are available starting with update 200769.*

```
; d s
Set   Items   Postings   Description
S1    823439   3988890   S CPU OR CPUS OR PROCESSOR? ? OR MICROPROCESSOR? ? OR
CENTRAL()PROCESSING()UNIT? ?
S2     21023     89346   S S1(3N)(USAGE OR UTILIZ? OR UTILIS? OR COST? ? OR LOAD???)
S3   3542302   13491476   S TASK? ? OR PROCESS?? OR JOB? ?
S4    128481    491778   S S3(3N)(CLASS?? OR CLASSIFICATION? ? OR CATEGOR? OR GROUP??? OR
SUBCLASS?? OR SUBGROUP? OR SUBCATEGOR?)
S5     96163    575952   S HIERARCH? OR TREE? ? OR BTREE? ?
S6   1166840   4812969   S WEIGH???? OR IMPORTANCE OR PRIORIT?
S7        62       390   S S2(3N)(MULTIPLY??? OR MULTIPLIE? ? OR MULTIPLICATION OR
CO()EFFICIENT? ? OR COEFFICIENT? ?)
S8         6        85   S S7 AND S6
S9        14       123   S S7 AND S4
S10       13       115   S S9 NOT S8
S11       11       104   S S10 NOT AD=20031128:20080102/PR
S12       40      1040   S S2 AND S4 AND S5
S13        4       135   S S12 AND S6
S14        4       135   S S13 NOT (S8 OR S11)
S15       22       495   S S2 AND S6(7N)S4
S16       22       495   S S15 NOT (S8 OR S11 OR S14)
S17       21       419   S S16 NOT AD=20031128:20080102/PR
S18      503      7917   S S2 AND WEIGH????
S19       27       799   S S18 AND S4
S20       20       349   S S19 NOT (S8 OR S11 OR S14 OR S16)
S21       20       349   S S20 NOT AD=20031128:20080102/PR
```

8/5/1 (Item 1 from file: 347) Links
JAPIO

04190011 **Image available**
SYSTEM FOR CHARGING EQUIPMENT NORMALITY CONFIRMING JOB

Pub. No.: 05-181711 [JP 5181711 A]
Published: July 23, 1993 (19930723)
Inventor: TANIMARU SENKICHI
Applicant: NEC CORP [000423] (A Japanese Company or Corporation), JP (Japan)
Application No.: 04-000026 [JP 9226]
Filed: January 06, 1992 (19920106)
International Class: [5] G06F-011/32; G06F-003/14; G06F-009/46
JAPIO Class: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units); 45.3 (INFORMATION PROCESSING -- Input Output Units)
Journal: Section: P, Section No. 1638, Vol. 17, No. 602, Pg. 93, November 05, 1993 (19931105)

ABSTRACT
PURPOSE: To automatically execute an apparatus normality confirming job at proper timing considering a CPU utilization rate and job **priority** order.

CONSTITUTION: An arithmetic operation part 24 monitors displays 104, 304 and **multiplies** the **CPU utilization** rate of a **central processing unit**(CPU) 54 by the job **priority** order at every constant interval of time. The multiplied result is applied to a comparing part 34 as a normal job coefficient. A timer 35 counts up real time and an adder 36 accumulatively adds a fixed value applied at every constant interval of time based upon the counted result of the timer 35 and applies the added result to the comparing part 34 as the equipment normality confirming job coefficient. The comparing part 34 compares the normal job coefficient applied from the operation part 24 with the equipment normality confirming job coefficient applied from the adder 36, and when the latter coefficient is larger than the former one, informs the CPU 54 of a previously registered equipment normality confirming job based upon a charging instruction 204. Thereby a job scheduler 74 sets up the **priority** order of the job to the highest **priority**.

17/5/1 (Item 1 from file: 347) Links
JAPIO

00718242 **Image available**
CONTROLLING METHOD FOR ENGINE

Pub. No.: 56-038542 [JP 56038542 A]
Published: April 13, 1981 (19810413)
Inventor: AMANO MATSUO
SUGAWARA TORU
IMAI MASUMI
FURUHASHI TOSHIO
Applicant: HITACHI LTD [000510] (A Japanese Company or Corporation), JP (Japan)
Application No.: 54-112953 [JP 79112953]
Filed: September 05, 1979 (19790905)
International Class: [3] F02D-035/00; F02D-005/00; G05B-015/02
JAPIO Class: 21.2 (ENGINES & TURBINES, PRIME MOVERS -- Internal Combustion); 22.3 (MACHINERY -- Control & Regulation); 45.4 (INFORMATION PROCESSING -- Computer Applications)
JAPIO Keyword: R131 (INFORMATION PROCESSING -- Microcomputers & Microprocessers)
Journal: Section: M, Section No. 73, Vol. 05, No. 90, Pg. 150, June 12, 1981 (19810612)

ABSTRACT
PURPOSE: To reduce the load factor of a micro-computer for controlling an engine by providing the computer with task groups divided into various control program units in the order of priority, and comparing and discriminating between the tasks on each level of the order.

CONSTITUTION: A program for controlling task groups comprises an initial processing program 202 performing the pre-edit of the computer operation, an interrupt processing program 206 analyzing various interrupt factors and demanding the start of necessary task group through a task dispatcher 208, and a macro-processing program 228 signalling tne end of run of the **tasks**. The **task groups** are provided with the **priority** order for each program used most frequently according to the engine operation conditions, and the dispatcher 208 assigns the occupation time and run time in a CPU according to the priority order in order to shorten the run time of the programs in the **CPU**, so that its **load** factor is reduced.

17/5/7 (Item 5 from file: 350) **Links**
Derwent WPIX

0013515409 *Drawing available*
WPI Acc no: 2003-608364/200357
XRPX Acc No: N2003-485053
Setting priority levels in multiple-programming computer system with priority scheduling uses escalation threshold to set priority levels
Patent Assignee: JANSSEN B (JANS-I); JANSSEN P G (JANS-I); REAL ENTERPRISE SOLUTIONS DEV BV (SOLU-N)
Inventor: JANSEN P G; JANSSEN B; JANSSEN P G

Patent Family (6 patents, 99 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	Type
WO 2003065214	A1	20030807	WO 2002NL70	A	20020130	200357	B
AU 2002230272	A1	20030902	AU 2002230272	A	20020130	200422	E
			WO 2002NL70	A	20020130		
EP 1474744	A1	20041110	EP 2002711528	A	20020130	200473	E
			WO 2002NL70	A	20020130		
NZ 534314	A	20050429	NZ 534314	A	20020130	200532	E
			WO 2002NL70	A	20020130		
JP 2005516303	W	20050602	WO 2002NL70	A	20020130	200541	E
			JP 2003564738	A	20020130		
US 20060037025	A1	20060216	WO 2002NL70	A	20020130	200614	E
			US 2005503408	A	20050208		

Priority Applications (no., kind, date): WO 2002NL70 A 20020130

Patent Details

Patent Number	Kind	Lan	Pgs	Draw	Filing Notes	
WO 2003065214	A1	EN	11	3		
National Designated States,Original	AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CO CR CU CZ DE DK DM DZ EC EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ OM PH PL PT RO RU SD SE SG SI SK SL TJ TM TN TR TT TZ UA UG US UZ VN YU ZA ZM ZW					
Regional Designated States,Original	AT BE CH CY DE DK EA ES FI FR GB GH GM GR IE IT KE LS LU MC MW MZ NL OA PT SD SE SL SZ TR TZ UG ZM ZW					
AU 2002230272	A1	EN			PCT Application	WO 2002NL70
					Based on OPI patent	WO 2003065214
EP 1474744	A1	EN			PCT Application	WO 2002NL70
					Based on OPI patent	WO 2003065214
Regional Designated States,Original	AL AT BE CH CY DE DK ES FI FR GB GR IE IT LI LT LU LV MC MK NL PT RO SE SI TR					
NZ 534314	A	EN			PCT Application	WO 2002NL70
					Based on OPI patent	WO 2003065214
JP 2005516303	W	JA	14		PCT Application	WO 2002NL70
					Based on OPI patent	WO 2003065214
US 20060037025	A1	EN			PCT Application	WO 2002NL70

Alerting Abstract WO A1
NOVELTY - The relative **usage** of a **CPU** is expressed as a percentage of the processing time and the escalation

threshold has been set at 80%. If the threshold is exceeded at a point To, the corresponding process is passed to a lower **process priority class**, to reduce the usage to about 95% and to allow a second process to proceed without much hindrance.
DESCRIPTION - INDEPENDENT CLAIMS are included for a multiple-programming computer system and for a computer program.
USE - Setting priority levels in multiple-programming computer system.
ADVANTAGE - Improves performance of the system.
DESCRIPTION OF DRAWINGS - The drawing shows a graph of relative **usage** of **CPU** processing power.

17/5/10 (Item 8 from file: 350) **Links**
Derwent WPIX

0012925578 *Drawing available*
WPI Acc no: 2003-002005/200301
XRPX Acc No: N2003-001481
Allocating percentage of system resources among process groups in a computer system comprising at least one CPU by allocating system resources of each of processor set to each of process groups according to number of assigned shares
Patent Assignee: SUN MICROSYSTEMS INC (SUNM)
Inventor: DOROFEEV A V; TUCKER A G

Patent Family (3 patents, 27 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	Type
EP 1253516	A2	20021030	EP 20029207	A	20020424	200301	B
US 20020161817	A1	20021031	US 2001843426	A	20010425	200301	E
US 7036123	B2	20060425	US 2001843426	A	20010425	200628	E

Priority Applications (no., kind, date): US 2001843426 A 20010425

Patent Details

Patent Number	Kind	Lan	Pgs	Draw	Filing Notes
EP 1253516	A2	EN	7	1	
Regional Designated States,Original	AL AT BE CH CY DE DK ES FI FR GB GR IE IT LI LT LU LV MC MK NL PT RO SE SI TR				

Alerting Abstract EP A2
NOVELTY - The method involves assigning each of process groups a number of shares for each or the at least one processor set. The system resources of each of the at least one processor set are allocated to each of the process groups according to the number of shares assigned to the each of the process groups based on a number of shares of all active groups within each of processor set.
DESCRIPTION - INDEPENDENT CLAIMS are included for:

A. a computer readable medium embodying a program for allocating a percentage of system resources among processes groups in a computer system
B. a scheduler for allocating a percentage of system resources among processes groups in a computer system

USE - For managing **load** of a **central processing unit** in a **computer** system and **also for allocating central processing unit** percentage **usage** among several processes **executing a specific** separate **set** of processes.
ADVANTAGE - Provides resource allocation scheme that would leverage advantages of the processor set architecture and an ability to distribute CPU resources among multiple process groups in a pre-defined manner.
DESCRIPTION OF DRAWINGS - The drawing illustrates exemplary allocation of system resources according to the inventive concept.

21/5/7 (Item 6 from file: 350) **Links**
Derwent WPIX

0012257214 *Drawing available*
WPI Acc no: 2002-197315/200226
XRPX Acc No: N2002-149884
Generalized processor sharing scheduler for communication network, adjusts initial connection weight by maintaining minimum quality of service level, based on output bandwidth used by each connection class
Patent Assignee: LUCENT TECHNOLOGIES INC (LUCE)
Inventor: KUMARAN K; MARGRAVE G E; MITRA D; STANLEY K R

Patent Family (1 patents, 26 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	Type
EP 1137316	A2	20010926	EP 2001302251	A	20010312	200226	B

Priority Applications (no., kind, date): US 2000531344 A 20000321

Patent Details

Patent Number	Kind	Lan	Pgs	Draw	Filing Notes
EP 1137316	A2	EN	34	11	
Regional Designated States,Original	AL AT BE CH CY DE DK ES FI FR GB GR IE IT LI LT LU LV MC MK NL PT RO SE SI TR				

Alerting Abstract EP A2
NOVELTY - A fraction of output bandwidth is allocated to each of input connections (110_1.1, 110K.1, 110_1.J, 110K.J) of each of connection classes (110_1, 110J) based on preset initial connection **weight**. The fraction of output bandwidth used by each **connection** class is measured, based on which initial connection **weight** is adjusted by maintaining minimum quality of service (**QoS**) level for each class.
DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

C. Generalized processor sharing (GPS) scheduling and network control method; Iterative method for choosing GPS **weights**;
D. Processor sharing node

USE - For regulating traffic in communication network.
ADVANTAGE - Allows substantial network sharing capacity as well as isolation and quality of service (QoS) guarantees by assigning **weight** to individual connections. Allows arbitrary number of **QoS** classes of service for an arbitrary number of class connection on a common truck.
DESCRIPTION OF DRAWINGS - The figure shows multiclass generalized processor sharing (GPS) scheduler.
110_1.1, 110K.1, 110_1.J, 110K.J Input connections
110_1, 110J Connection classes